

Exhibit I

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)Java™ 2 Platform
Std. Ed. v1.4.2[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

java.security

Class Security

[java.lang.Object](#)└─ [java.security.Security](#)public final class **Security**
extends [Object](#)

This class centralizes all security properties and common security methods. One of its primary uses is to manage providers.

Method Summary

static int	addProvider (Provider provider) Adds a provider to the next position available.
static String	getAlgorithmProperty (String algName, String propName) Deprecated. <i>This method used to return the value of a proprietary property in the master file of the "SUN" Cryptographic Service Provider in order to determine how to parse algorithm-specific parameters. Use the new provider-based and algorithm-independent AlgorithmParameters and KeyFactory engine classes (introduced in the Java 2 platform) instead.</i>
static Set	getAlgorithms (String serviceName) Returns a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore).
static String	getProperty (String key) Gets a security property value.
static Provider	getProvider (String name) Returns the provider installed with the specified name, if any.
static Provider []	getProviders () Returns an array containing all the installed providers.
static Provider []	getProviders (Map filter) Returns an array containing all installed providers that satisfy the specified selection criteria, or null if no such providers have been installed.
static Provider []	getProviders (String filter) Returns an array containing all installed providers that satisfy the specified

	selection criterion, or null if no such providers have been installed.
static int	insertProviderAt (Provider provider, int position) Adds a new provider, at a specified position.
static void	removeProvider (String name) Removes the provider with the specified name.
static void	setProperty (String key, String datum) Sets a security property value.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Method Detail

getAlgorithmProperty

```
public static String getAlgorithmProperty(String algName,
                                           String propName)
```

Deprecated. *This method used to return the value of a proprietary property in the master file of the "SUN" Cryptographic Service Provider in order to determine how to parse algorithm-specific parameters. Use the new provider-based and algorithm-independent [AlgorithmParameters](#) and [KeyFactory](#) engine classes (introduced in the Java 2 platform) instead.*

Gets a specified property for an algorithm. The algorithm name should be a standard name. See Appendix A in the [Java Cryptography Architecture API Specification & Reference](#) for information about standard algorithm names. One possible use is by specialized algorithm parsers, which may map classes to algorithms which they understand (much like Key parsers do).

Parameters:

algName - the algorithm name.
propName - the name of the property to get.

Returns:

the value of the specified property.

insertProviderAt

```
public static int insertProviderAt(Provider provider,
                                   int position)
```

Adds a new provider, at a specified position. The position is the preference order in which providers are searched for requested algorithms. Note that it is not guaranteed that this preference will be respected. The position is 1-based, that is, 1 is most preferred, followed by 2, and so on.

If the given provider is installed at the requested position, the provider that used to be at that position, and all providers with a position greater than `position`, are shifted up one position (towards the end of the list of installed providers).

A provider cannot be added if it is already installed.

First, if there is a security manager, its `checkSecurityAccess` method is called with the string `"insertProvider."+provider.getName()` to see if it's ok to add a new provider. If the default implementation of `checkSecurityAccess` is used (i.e., that method is not overridden), then this will result in a call to the security manager's `checkPermission` method with a `SecurityPermission("insertProvider."+provider.getName())` permission.

Parameters:

`provider` - the provider to be added.

`position` - the preference position that the caller would like for this provider.

Returns:

the actual preference position in which the provider was added, or -1 if the provider was not added because it is already installed.

Throws:

[SecurityException](#) - if a security manager exists and its

[SecurityManager.checkSecurityAccess\(java.lang.String\)](#) method denies access to add a new provider

See Also:

[getProvider\(java.lang.String\)](#), [removeProvider\(java.lang.String\)](#), [SecurityPermission](#)

addProvider

```
public static int addProvider(Provider provider)
```

Adds a provider to the next position available.

First, if there is a security manager, its `checkSecurityAccess` method is called with the string `"insertProvider."+provider.getName()` to see if it's ok to add a new provider. If the default implementation of `checkSecurityAccess` is used (i.e., that method is not overridden), then this will result in a call to the security manager's `checkPermission` method with a `SecurityPermission("insertProvider."+provider.getName())` permission.

Parameters:

`provider` - the provider to be added.

Returns:

the preference position in which the provider was added, or -1 if the provider was not added because it is already installed.

Throws:

[SecurityException](#) - if a security manager exists and its

[SecurityManager.checkSecurityAccess\(java.lang.String\)](#) method denies access to add a new provider

See Also:

[getProvider\(java.lang.String\)](#), [removeProvider\(java.lang.String\)](#),
[SecurityPermission](#)

removeProvider

```
public static void removeProvider(String name)
```

Removes the provider with the specified name.

When the specified provider is removed, all providers located at a position greater than where the specified provider was are shifted down one position (towards the head of the list of installed providers).

This method returns silently if the provider is not installed.

First, if there is a security manager, its `checkSecurityAccess` method is called with the string `"removeProvider."+name` to see if it's ok to remove the provider. If the default implementation of `checkSecurityAccess` is used (i.e., that method is not overridden), then this will result in a call to the security manager's `checkPermission` method with a `SecurityPermission` (`"removeProvider."+name`) permission.

Parameters:

name - the name of the provider to remove.

Throws:

[SecurityException](#) - if a security manager exists and its [SecurityManager.checkSecurityAccess\(java.lang.String\)](#) method denies access to remove the provider

See Also:

[getProvider\(java.lang.String\)](#), [addProvider\(java.security.Provider\)](#)

getProviders

```
public static Provider[] getProviders()
```

Returns an array containing all the installed providers. The order of the providers in the array is their preference order.

Returns:

an array of all the installed providers.

getProvider

```
public static Provider getProvider(String name)
```

Returns the provider installed with the specified name, if any. Returns null if no provider with the specified name is installed.

Parameters:

name - the name of the provider to get.

Returns:

the provider of the specified name.

See Also:

[removeProvider\(java.lang.String\)](#), [addProvider\(java.security.Provider\)](#)

getProviders

```
public static Provider[] getProviders(String filter)
```

Returns an array containing all installed providers that satisfy the specified selection criterion, or null if no such providers have been installed. The returned providers are ordered according to their [preference order](#).

A cryptographic service is always associated with a particular algorithm or type. For example, a digital signature service is always associated with a particular algorithm (e.g., DSA), and a CertificateFactory service is always associated with a particular certificate type (e.g., X.509).

The selection criterion must be specified in one of the following two formats:

- *<crypto_service>.<algorithm_or_type>*

The cryptographic service name must not contain any dots.

A provider satisfies the specified selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service.

For example, "CertificateFactory.X.509" would be satisfied by any provider that supplied a CertificateFactory implementation for X.509 certificates.

- *<crypto_service>.<algorithm_or_type> <attribute_name>:< attribute_value>*

The cryptographic service name must not contain any dots. There must be one or more space characters between the *<algorithm_or_type>* and the *<attribute_name>*.

A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service and its implementation meets the constraint expressed by the specified attribute name/value pair.

For example, "Signature.SHA1withDSA KeySize:1024" would be satisfied by any provider that implemented the SHA1withDSA signature algorithm with a keysize of 1024 (or larger).

See Appendix A in the [Java Cryptography Architecture API Specification & Reference](#) for information about standard cryptographic service names, standard algorithm names and standard attribute names.

Parameters:

filter - the criterion for selecting providers. The filter is case-insensitive.

Returns:

all the installed providers that satisfy the selection criterion, or null if no such providers have been installed.

Throws:

[InvalidParameterException](#) - if the filter is not in the required format

See Also:

[getProviders\(java.util.Map\)](#)

getProviders

```
public static Provider[] getProviders(Map filter)
```

Returns an array containing all installed providers that satisfy the specified selection criteria, or null if no such providers have been installed. The returned providers are ordered according to their [preference order](#).

The selection criteria are represented by a map. Each map entry represents a selection criterion. A provider is selected iff it satisfies all selection criteria. The key for any entry in such a map must be in one of the following two formats:

- *<crypto_service>.<algorithm_or_type>*

The cryptographic service name must not contain any dots.

The value associated with the key must be an empty string.

A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service.

- *<crypto_service>.<algorithm_or_type> <attribute_name>*

The cryptographic service name must not contain any dots. There must be one or more space characters between the *<algorithm_or_type>* and the *<attribute_name>*.

The value associated with the key must be a non-empty string. A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service and its implementation meets the constraint expressed by the specified attribute name/value pair.

See Appendix A in the [Java Cryptography Architecture API Specification & Reference](#) for information about standard cryptographic service names, standard algorithm names and standard attribute names.

Parameters:

filter - the criteria for selecting providers. The filter is case-insensitive.

Returns:

all the installed providers that satisfy the selection criteria, or null if no such providers have

been installed.

Throws:

[InvalidParameterException](#) - if the filter is not in the required format

See Also:

[getProviders\(java.lang.String\)](#)

getProperty

```
public static String getProperty(String key)
```

Gets a security property value.

First, if there is a security manager, its `checkPermission` method is called with a `java.security.SecurityPermission("getProperty."+key)` permission to see if it's ok to retrieve the specified security property value..

Parameters:

`key` - the key of the property being retrieved.

Returns:

the value of the security property corresponding to `key`.

Throws:

[SecurityException](#) - if a security manager exists and its [SecurityManager.checkPermission\(java.security.Permission\)](#) method denies access to retrieve the specified security property value

See Also:

[setProperty\(java.lang.String, java.lang.String\)](#), [SecurityPermission](#)

setProperty

```
public static void setProperty(String key,  
                               String datum)
```

Sets a security property value.

First, if there is a security manager, its `checkPermission` method is called with a `java.security.SecurityPermission("setProperty."+key)` permission to see if it's ok to set the specified security property value.

Parameters:

`key` - the name of the property to be set.

`datum` - the value of the property to be set.

Throws:

[SecurityException](#) - if a security manager exists and its [SecurityManager.checkPermission\(java.security.Permission\)](#) method denies access to set the specified security property value

See Also:

[getProperty\(java.lang.String\)](#), [SecurityPermission](#)

getAlgorithms

```
public static Set getAlgorithms(String serviceName)
```

Returns a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore). Returns an empty Set if there is no provider that supports the specified service. For a complete list of Java cryptographic services, please see the [Java Cryptography Architecture API Specification & Reference](#). Note: the returned set is immutable.

Parameters:

`serviceName` - the name of the Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore). Note: this parameter is case-insensitive.

Returns:

a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service or an empty set if no provider supports the specified service.

Since:

1.4

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

*Java™ 2 Platform
Std. Ed. v1.4.2*

[Submit a bug or feature](#)

For further API reference and developer documentation, see [Java 2 SDK SE Developer Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright © 2003, 2010 Oracle and/or its affiliates. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#).



public final class

Security

extends [Object](#)[java.lang.Object](#)

↳ java.security.Security

Class Overview

`Security` is the central class in the Java Security API. It manages the list of security `Provider` that have been installed into this runtime environment.

Summary

Public Methods	
static int	<code>addProvider (Provider provider)</code> Adds the given <code>provider</code> to the collection of providers at the next available position.
static String	<code>getAlgorithmProperty (String algName, String propName)</code> <i>This method is deprecated. Use <code>AlgorithmParameters</code> and <code>KeyFactory</code> instead.</i>
static Set<String>	<code>getAlgorithms (String serviceName)</code> Returns a <code>Set</code> of all registered algorithms for the specified cryptographic service.
static String	<code>getProperty (String key)</code> Returns the value of the security property named by the argument.
synchronized static Provider	<code>getProvider (String name)</code> Returns the <code>Provider</code> with the specified name.
synchronized static Provider[]	<code>getProviders (Map<String, String> filter)</code> Returns the array of providers which meet the user supplied set of filters.
static Provider[]	<code>getProviders (String filter)</code> Returns the array of providers which meet the user supplied string filter.
synchronized static Provider[]	<code>getProviders ()</code> Returns an array containing all installed providers.
synchronized static int	<code>insertProviderAt (Provider provider, int position)</code> Insert the given <code>Provider</code> at the specified <code>position</code> .
synchronized static void	<code>removeProvider (String name)</code> Removes the <code>Provider</code> with the specified name form the collection of providers.
static void	<code>setProperty (String key, String datum)</code> Sets the value of the specified security property.

Inherited Methods

[\[Expand\]](#)▶ From class [java.lang.Object](#)

Public Methods

```
public static int addProvider (Provider provider)
```

Since: API Level 1

Adds the given `provider` to the collection of providers at the next available position.

If a `SecurityManager` is installed, code calling this method needs the `SecurityPermission` `insertProvider.NAME` (where `NAME` is the provider name) to be granted, otherwise a `SecurityException` will be thrown.

Parameters

`provider` the provider to be added.

Returns

the actual position or `-1` if the given `provider` was already in the list.

Throws

[SecurityException](#) if a `SecurityManager` is installed and the caller does not have permission to invoke this method.

```
public static String getAlgorithmProperty (String algName, String propName)
```

Since: API Level 1

This method is deprecated.

Use [AlgorithmParameters](#) and [KeyFactory](#) instead.

Returns value for the specified algorithm with the specified name.

Parameters

`algName` the name of the algorithm.

`propName` the name of the property.

Returns

value of the property.

```
public static Set<String> getAlgorithms (String serviceName)
```

Since: API Level 1

Returns a `Set` of all registered algorithms for the specified cryptographic service. "`Signature`", "`Cipher`" and "`KeyStore`" are examples for such kind of services.

Parameters

`serviceName` the case-insensitive name of the service.

Returns

a `Set` of all registered algorithms for the specified cryptographic service, or an empty `Set` if `serviceName` is `null` or if no registered provider provides the requested service.

```
public static String getProperty (String key)
```

Since: API Level 1

Returns the value of the security property named by the argument.

If a `SecurityManager` is installed, code calling this method needs the `SecurityPermission` `getProperty.KEY`

(where KEY is the specified `key`) to be granted, otherwise a `SecurityException` will be thrown.

Parameters

`key` the name of the requested security property.

Returns

the value of the security property.

Throws

[SecurityException](#) if a `SecurityManager` is installed and the caller does not have permission to invoke this method.

public static synchronized [Provider](#) **getProvider** ([String](#) name)

Since: API Level 1

Returns the `Provider` with the specified name. Returns `null` if name is `null` or no provider with the specified name is installed.

Parameters

`name` the name of the requested provider.

Returns

the provider with the specified name, maybe `null`.

public static synchronized [Provider\[\]](#) **getProviders** ([Map](#)<[String](#), [String](#)> filter)

Since: API Level 1

Returns the array of providers which meet the user supplied set of filters. The filter must be supplied in one of two formats:

- CRYPTO_SERVICE_NAME.ALGORITHM_OR_TYPE

for example: "MessageDigest.SHA" The value associated with the key must be an empty string.

- CRYPTO_SERVICE_NAME.ALGORITHM_OR_TYPE ATTR_NAME:ATTR_VALUE

for example: "Signature.MD2withRSA KeySize:512" where "KeySize:512" is the value of the filter map entry.

Parameters

`filter` case-insensitive filter.

Returns

the providers which meet the user supplied string filter `filter`. A `null` value signifies that none of the installed providers meets the filter specification.

Throws

[InvalidParameterException](#) if an unusable filter is supplied.

[NullPointerException](#) if `filter` is `null`.

public static [Provider\[\]](#) **getProviders** ([String](#) filter)

Since: API Level 1

Returns the array of providers which meet the user supplied string filter. The specified filter must be supplied in one of two formats:

- CRYPTO_SERVICE_NAME.ALGORITHM_OR_TYPE

(for example: "MessageDigest.SHA")

- CRYPTO_SERVICE_NAME.ALGORITHM_OR_TYPE ATTR_NAME:ATTR_VALUE

(for example: "Signature.MD2withRSA KeySize:512")

Parameters

`filter` case-insensitive filter.

Returns

the providers which meet the user supplied string filter `filter`. A `null` value signifies that none of the installed providers meets the filter specification.

Throws

[`InvalidParameterException`](#) if an unusable filter is supplied.
[`NullPointerException`](#) if `filter` is `null`.

```
public static synchronized Provider\[\] getProviders ()
```

Since: API Level 1

Returns an array containing all installed providers. The providers are ordered according their preference order.

Returns

an array containing all installed providers.

```
public static synchronized int insertProviderAt (Provider provider, int position)
```

Since: API Level 1

Insert the given `Provider` at the specified `position`. The positions define the preference order in which providers are searched for requested algorithms.

If a `SecurityManager` is installed, code calling this method needs the `SecurityPermission` `insertProvider.NAME` (where NAME is the provider name) to be granted, otherwise a `SecurityException` will be thrown.

Parameters

`provider` the provider to insert.
`position` the position (starting from 1).

Returns

the actual position or `-1` if the given `provider` was already in the list. The actual position may be different from the desired position.

Throws

[`SecurityException`](#) if a `SecurityManager` is installed and the caller does not have permission to invoke this method.

```
public static synchronized void removeProvider (String name)
```

Since: API Level 1

Removes the `Provider` with the specified name form the collection of providers. If the the `Provider` with the specified name is removed, all provider at a greater position are shifted down one position.

Returns silently if `name` is `null` or no provider with the specified name is installed.

If a `SecurityManager` is installed, code calling this method needs the `SecurityPermission` `removeProvider.NAME` (where NAME is the provider name) to be granted, otherwise a `SecurityException` will be thrown.

Parameters

`name` the name of the provider to remove.

Throws

[`SecurityException`](#) if a `SecurityManager` is installed and the caller does not have permission to invoke this method.

```
public static void setProperty (String key, String datum)
```

Since: API Level 1

Sets the value of the specified security property.

If a `SecurityManager` is installed, code calling this method needs the `SecurityPermission` `setProperty.KEY`

(where KEY is the specified `key`) to be granted, otherwise a `SecurityException` will be thrown.

Parameters

`key` the name of the security property.
`datum` the value of the security property.

Throws

[SecurityException](#) if a `SecurityManager` is installed and the caller does not have permission to invoke this method.

Except as noted, this content is licensed under [Apache 2.0](#). For details and restrictions, see the [Content License](#).
Android 2.2 r1 - 12 Oct 2010 1:21

[Site Terms of Service](#) - [Privacy Policy](#) - [Brand Guidelines](#)